VZ LINK magazine

Fifth Anniversary

OCTOBER 1991 **ISSUE 60**

NEW ZEALAND

VZ 300/200
COMPUTER USERS CLUB

EDITOR'S COMMENT

It is hard to believe that this is the Fifth Anniversary of VZ LINK and the New Zealand VZ users Club.
It is always hard work putting the newsletter together. I really enjoy the challenge and thou I still have a few problems with layout and spelling errors, I would like to think that the newsletter was improving.
While on the subject of improvements, I have been given a few suggestions re layout and you should see a slightly different format next year.

It takes the whole club to produce this newsletter, without your programs and other input it would be impossible.
As you can see by this issues cover I am still pleading for programs etc for inclusion in the newsletter. It doesn't have to be flash but it must work and must have instructions. So once again I beg. !!!!!!!!
Please send in those programs and items that you have written. I can't produce a newsletter without your input.

There are still a lot of out standing subscriptions due. If yours is one of them please forward it as soon as possible. The club relies on these to continue the newsletters. If you are not sure when and if yours is due contact me and I will tell you. This issue will be your last if you are overdue.

REMEMBER

If you want to contact me about any problems the phone number is now: Ph. 520-6743
If it is really urgent you can contact me at work, the number is now: Ph. 270-5389

CLUB MEETING

The last club meeting for the year will be held on :-

>>>>>>>>>>>>>>>>>>>> SATURDAY 30 NOVEMBER 1991 <<<<<<<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>>> Starting at 10.30 am. <<<<<<<<<<<<<<<<<<<<<<<<<

This date is confirmed, please mark it on you calendar.
The Venue will be the same as in the past. See the insert for more details.

AT THE MEETING

That the club meeting we hope to have the following :-
Peter Patterson will be fitting reset buttons to VZ's for those that require it.
Laurie Whetton will be checking Disk Drive alignments for those that need it.

FOR SALE

Steelo Lemon soap pads, box of 10 for $1.00
Cassette Leads approx 1 metre long for $7.00 each (suppled by Ken Clarke)
Disks for $1.00 each /Cassettes for $2.00 each.
All proceeds to club funds.

WANTED

There are one or two members who need printer interfaces at the moment.
If you have a spare one available for sale, please let me know.
Maybe someone will offer to build up a couple, who knows.!

$$$

DON'T TRASH YOUR DISKS!

Dust and Other Particles

Just like flies at a picnic, dust seems to descend on computer equipment. Interestingly, the dust is attracted to the display monitor in the same manner as it is to a television screen. If the dust isn't cleaned from the screen, it will build up, and eventually someone will rub it and mar the screen surface.

The static electricity charge that builds up in the computer and the display monitor attracts dust and dirt. That's why large computer systems are kept in cool, clean computer rooms. They require special air conditioning and dust-free spaces because the large equipment generates more heat and is just as susceptible to failures caused by dust build-up.

Dust and dirt build-up insulates the circuit devices and prevents the release of the heat generated during normal operation. If the devices can't dissipate this heat, the inside temperature rises higher than normal, causing the chips and other components to wear out even faster. Dust is a major contributor to memory chip failure. Dust seems to be attracted to heat. Have you ever noticed that dust builds up on light bulbs in your lamps or on the tops of stereos and televisions more than it does on cooler objects? The dust particles are charged and are attracted to the magnetic field around electrical equipment. Your computer system problems increase in direct proportion to the increase in dust buildup.

Mechanical devices such as printers and disk drives fail more often than solid-state electronic devices because mechanical and electromechanical devices have moving parts that get dirty easily, causing overheating and earlier failure. Look inside your printer and you'll see the kinds of dirt and dust that are collecting. Paper sheds tiny particles as it moves through the printer. These particles become insulators to prevent the heat generated during normal operation from escaping into the air.

Disk drives have more dust-related problems than printers because they are designed with read/write heads that operate on or slightly above the diskette. The space between the head and the disk is small. When the head rides on the disk surface, dust and dirt can cause major problems (Fig. 1).

Foreign particles such as dirt, smoke ash, and tiny fibers can cause catastrophic problems in diskette jackets and in disk drives themselves. The air we breathe is full of airborne particles, but most of these are too small to even be seen, let alone become a problem. The larger particles in the air cause computer system problems. Cigarette ash, for example, can settle on a disk surface and move from track to track inside the disk jacket, causing loss of data.

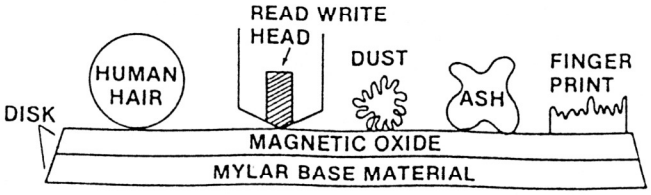


Fig. 1. With a read-head that rides on the surface of the disk, any small piece of foreign material can cause problems.

Inside the vinyl jacket surrounding each of your disks is a special lining that traps dirt and dust as the disk spins in the drive. This doesn't mean you can get careless about dust and dirt. Dirt on a disk can be swept off by the drive read/write head and gouge out a path on the disk surface. Or it can stick on the head and cause other disks to be gouged. The dirt can also cause the head itself to corrode and wear out.

Smoke from cigarettes and cigars can coat the internal surfaces of the disk drive with a gummy soot that not only can produce data transfer errors, but can also interfere with the mechanical operation, further increasing the wear on the drive. Tobacco is also believed to cause rapid oxidation on pins and connectors, increasing the likelihood of intermittent errors. Most computer centers and computer rooms are off-limits for smoking.

Dust buildup can be controlled. Thoroughly cleaning your computer area every week will do much to keep your system in top condition. Dirt and dust can be removed from the equipment housings using a damp cloth lightly coated with mild soap. Clean electrical equipment with the power turned off. Be careful you don't wet or moisten the electronic components.

After washing the surface, I rewipe the outside of the equipment with a soft cloth dampened with a mixture of one part liquid fabric softener to three parts water. The chemical makeup of some liquid softeners is almost the same as antistatic chemical spray. Wiping the case and screen helps to keep static charges from attracting dust to the screen and tops of the hardware. The fabric softener is antimagnetic and prevents the attraction of dust. The chemicals in this inexpensive solution last longer than some antistatic sprays and help make your screens less susceptible to scratching.

Another quite successful technique is blowing dust away from the screen with a pressurized can of antistatic dusting spray, as shown in Fig. 2. Using this kind of product means you don't have to wipe your equipment off first. Wiping a screen should be done carefully, because you could scratch the screen if some hard dust or dirt particles are on the screen or your cloth.

ARTICLE SUBMITTED

BY KEN CLARKE.

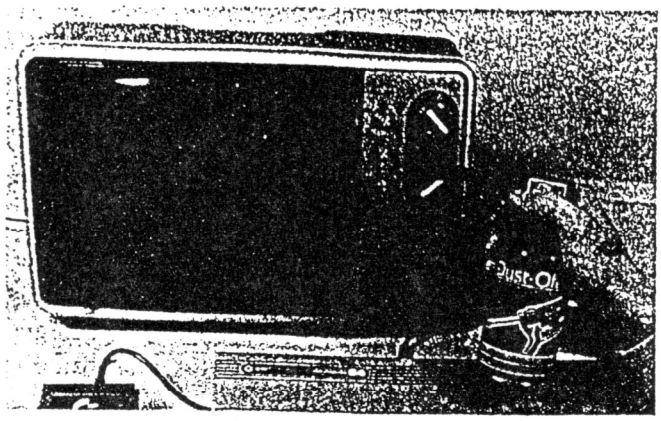


Fig. 2. An antistatic spray for computer equipment.

The following are some manufacturer-recommended cleaning methods:

Use one part fabric softener to three parts water to clean your screen.

Use mild soap and water, with a soft cloth for drying.

Use a window cleaner spray.

Note: Although the monitor literature from several manufacturers recommends this, be careful. Common household aerosol sprays, solvents, polishes, or cleaning agents may damage your monitor cabinet and screen. The safest cleaning solution is mild soap and water.

Another way to keep dust off is to use an antistatic spray.

Associated with cleaning advice, each manufacturer also includes an important safety precaution:

Caution: Make *sure* the power is off and the plug(s) pulled out of the power socket(s). Use a damp cloth. Don't let any liquid run or get into your equipment.

You can use a long plastic nozzle on the end of your vacuum hose to reach in and around everything inside the hardware. Dust and small particles can be cleaned off the circuit board inside your computer using a soft brush. Be careful you don't damage any of the parts. Brush lightly.

Another control measure is the use of dust covers. You may not have an air-conditioned, air-purified room in which to use your computer so dust covers become of paramount importance. Plastic covers, made static-free with an antistatic aerosol or by wiping the surface with the fabric softener-water mixture, will provide good dust protection for your system.

Here is a summary of ways to counter dust

Use dust covers.

Keep windows closed.

No smoking near your computer system.

No crumb-producing foods near your computer.

No liquids on any equipment.

Don't touch the surface of any floppy disk.

Vacuum the system and the area weekly.

Clean your monitor screen with static-reducing material.

ISLAND OF SECRETS PART 2

Adapted to the VZ by Pete Patterson

INSTRUCTIONS

The program is designed to run on a VZ300 or 200 with the Extended Basic ROM fitted,or alternately having an Extended Basic Utility loaded.

The VZ in most cases needs two-word commands such as GET WATER. There are special one word commands for Save ,Load and Quit.

The "TIME REMAINING",shown top right of your screen,constantly decreases. The game is over when the time is up. Typing INFO will show you what you are carrying. You start with two units of food and drink and you can add to this by picking up food and drink you spot during the course of the game. You can see that your strength and wisdom vary as the game progresses. Strength can be increased by eating and drinking. Wisdom can be increased only by doing the right things in the game.

You will need to refer constantly to the map to find your way around. At any point during the game you can stop and save your progress to disk,by typing XSAVE. When you want to continue,LOAD,RUN the game and XLOAD on the first instruction. If you give up, type QUIT and the game will end with your score given.

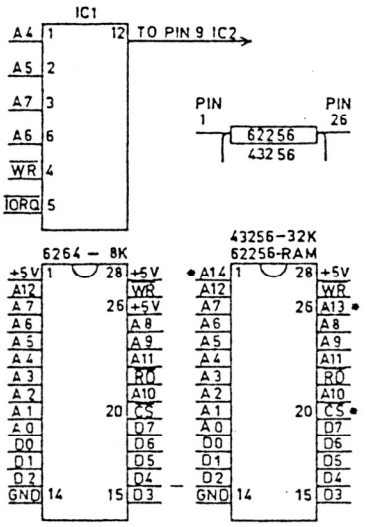
```
5 REM"[illegible]
10 REM ADAPTED TO THE VZ300 BY  PETE PATTERSON
12 REM"USE A VZ WITH UPDATED       EPROM FITTED.
13 REM"OR LOAD AN EXTENDED BASIC   BEFORE LOADING PROGRAM.
15 CLS:CLEAR1000
20 GOSUB2820
30 D=R:IFR=20,D=RND(80)
40 GOSUB650:GOSUB2770:PRINT"ISLAND OF SECRETS",
45 PRINT"TIME REMAINING:";L
50 PRINTG$;TAB(0);"STRENGTH = ";INT(Y);TAB(18);"WISDOM = ";X
55 PRINTG$
60 PRINT"YOU ARE ";I$(VAL(LEFT$(A$,1)));" ";:GOSUB720:N=0
70 FORI=1TOC4
80 C=0:READY$
90 IFL(I)=RANDF(I)<1,C=1
100 IFN>0ANDC=1,A$=A$+","
110 IFC=1,A$=A$+" "+Y$:N=N+1
120 NEXTI
130 IFN>0,A$="*YOU SEE"+A$:GOSUB720
140 PRINT:PRINTG$;:PRINTF$
150 PRINT:PRINT"WHAT WILL YOU DO";
160 INPUTE$
170 C$="":X$="":A=0:O=52:LI=LEN(E$)
180 FORI=1TOLI
190 IFMID$(E$,I,1)=" "ANDC$="",C$=LEFT$(E$,I-1)
200 IFMID$(E$,I+1,1)<>" "ANDC$>"",X$=RIGHT$(E$,LI-I):I=LI
210 NEXTI
220 IFX$="",C$=E$
230 IFLEN(C$)<3,C$=C$+"???"
240 FORI=1TOV
250 IFLEFT$(C$,3)=MID$(V$,3*(I-1)+1,3),A=I
260 NEXTI
270 GOSUB760
280 B$="":IFA=0,A=V+1
290 IFX$="???",F$="MOST ACTIONS NEED TWO WORDS"
300 IFA>VORO=52,F$=W$+C$+" "+X$
310 IFA>VANDO=52,F$="WHAT!"
320 L=L-1:Y=Y-(E/C4+.1)
330 B$=STR$(O)+STR$(L(O))+STR$(F(O))+STR$(R)
335 B$=STR$(VAL(B$)):B$=RIGHT$(B$,LEN(B$)-1)
340 ONINT(A/10)+1GOSUB590,600,610,620,630,630,640
350 IFR=61,X=X-RND(2)+1
360 IFR=14ANDRND(3)=1,Y=Y-1:F$="YOU ARE BITTEN"
370 IFF(36)<1AND-R<>F(22),F(36)=F(36)+1:L(36)=R:Y=Y-1
380 IFR<>L(16)ANDL(16)>0,L(16)=1+RND(4)
390 IFR<>L(39),L(39)=10*(RND(5)-1)+7+RND(3)
400 IFR=L(39)ANDR<>L(43)ANDF(13)>-1,Y=Y-2:X=X-2
410 IFR<78,L(32)=76+RND(2)
420 IFR=33ORR=57ORR=73ANDRND(2)=1,L(25)=R
430 IFR=L(32)ANDRND(2)=1ANDF(32)=0,GOSUB1310
440 IFR=19ANDY<70ANDF(43)=0ANDRND(4)=1,SL=1
445 IFSL=1,F$="PUSHED INTO THE PIT":F(W)=1:SL=0
450 IFR<>L(41),L(41)=21+(RND(3)*10)+RND(2)
460 IFR=L(41),F(41)=F(41)-1:IFF(41)<-4,GOSUB1230
470 IFF(43)=0,L(43)=R
480 IFL(43)<18ANDR<>9ANDR<>10ANDF(W-2)<1,GOSUB1330
490 IFR=18,Y=Y-1
500 IFY<50,O=RND(9):GOSUB1530:IFL(O)=R,F$="YOU DROP SOMETHING"
510 IFL<900ANDR=23ANDF(36)>0ANDRND(3)=3,GOSUB1360
520 IFR=47ANDF(8)>0,F$=F$+" YOU CAN GO NO FURTHER"
530 IFF(8)+F(11)+F(13)=-3,F(W)=1:GOSUB2800
540 IFF(W)=0ANDL>0ANDY>1ANDX>1,30
550 IFL<1ORY<1,F$="YOU HAVE FAILED,THE EVIL ONE SUCCEEDS"
560 PRINT:PRINTF$
```

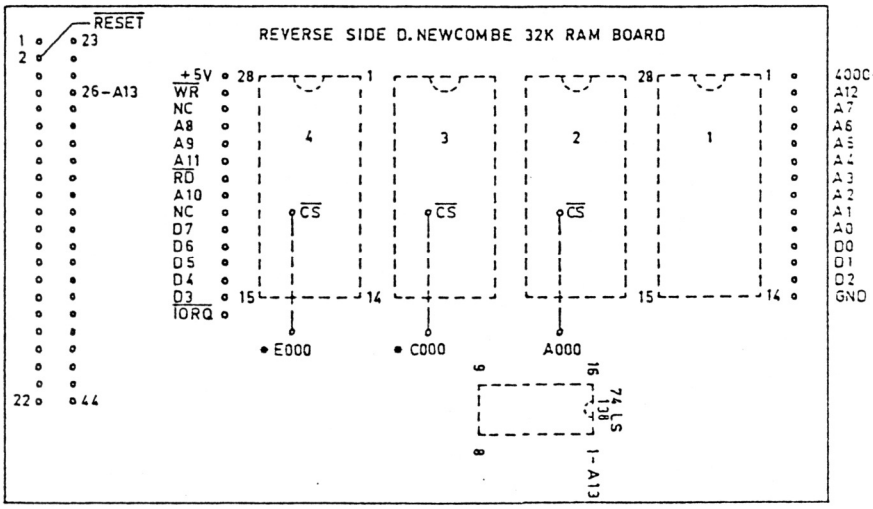
```
565 PRINT"YOUR FINAL SCORE=";INT(X+Y+(ABS(L/7*(L<640))))
570 PRINT:PRINT:PRINT"GAME OVER"
580 END
590 ONAGOSUB810,810,810,810,810,1080,1080,1390,1530:RETURN
600 ONA-9GOSUB1540,1630,1670,1710,1730,1080,1760,1760,1760,1760
605 RETURN
610 ONA-19GOSUB1820,1820,1820,1820,1910,2100,2200,2270,2270,1080
615 RETURN
620 ONA-29GOSUB2500,2500,2300,2300,2330,2350,2400,2400,2470,2540
625 RETURN
630 ONA-39GOSUB2600,2600,2720,640
640 RETURN
650 RESTORE
660 FORI=1TOD:READA$:NEXTI
670 REM
680 D$=RIGHT$(A$,4):A$=LEFT$(A$,LEN(A$)-4)
690 IFR=39,D$=MID$("101110100",RND(5),4)
700 IFR=20,D$="1110"
710 GOSUB2780:RETURN
720 FORI=2TOLEN(A$):E$=MID$(A$,I,1)
730 PRINTE$;:IFE$=" ",PRINT;
740 NEXTI
750 PRINT". ";:A$="":RETURN
760 IFLEN(X$)<3,X$=X$+"???"
770 FORI=1TOW
780 IFLEFT$(X$,3)=MID$(Z$,3*(I-1)+1,3),O=I
790 NEXTI:IFO=0,O=52
800 RETURN
810 D=0:C=0:IFO=52,D=A
820 IFO>C4ANDO<W,D=O-C4
830 IFB$="500012"ORB$="500053"ORB$="500045",D=4
840 IFB$="500070"ORB$="500037"ORB$="510011"ORB$="510041",D=1
850 IFB$="510043"ORB$="490066"ORB$="490051",D=1
860 IFB$="510060"ORB$="480056",D=2
870 IFB$="510044"ORB$="510052",D=3
880 IFB$="490051"ANDF(29)=0,GOSUB2110:RETURN
890 IFR=L(39)AND(X+Y<180ORR=10),F$=W$+"LEAVE!":RETURN
900 IFR=L(32)ANDF(32)<1ANDD=3,SL=1
905 IFSL=1,F$="HE WILL NOT LET YOU PAST":SL=0:RETURN
910 IFR=47ANDF(44)=0,F$="THE ROCKS MOVE TO PREVENT YOU":RETURN
920 IFR=28ANDF(7)<>1,F$="THE ARM HOLDS YOU FAST":RETURN
930 IFR=45ANDF(40)=0ANDD=4,F$="HISSSS!":RETURN
940 IFR=25ANDF(16)+L(16)<>-1ANDD=3THENSL=1
945 IFSL=1,F$="TOO STEEP TO CLIMB":SL=0:RETURN
950 IFR=51ANDD=3THENF$="THE DOOR IS BARRED!":RETURN
960 IFD>0,IFMID$(D$,D,1)="0",SL=1:C=1
965 IFSL=1,R=R+VAL(MID$("-10+10+01-01",D*3-2,3)):SL=0
970 F$="OK"
980 IFD<1ORC=0,F$=W$+"GO THAT WAY"
990 IFR=33ANDL(16)=0,SL=1
995 IFSL=1,L(16)=RND(4):F(16)=0:F$="THE BEAST RUNS AWAY!":SL=0
1000 IFR<>L(25)ORO<>25,RETURN
1010 F$="":A$="#YOU BOARD THE CRAFT "
1020 IFX<60,A$=A$+S$
1030 A$=A$+T$
1040 GOSUB2740:GOSUB2760:GOSUB2760
1050 IFX<60,A$="#TO SERVE OMEGAN FOREVER!":F(W)=1
1060 IFX>59,A$="#THE BOAT SKIMS THE DARK SILENT WATERS":R=57
1070 GOSUB2750:GOSUB2760:GOSUB2760:RETURN
1080 IF((F(O)>0ANDF(O)<9)ORL(O)<>R)ANDO<=C3,SL=1
1085 IFSL=1,F$="WHAT "+X$+"?":SL=0:RETURN
1090 IFB$="3450050",Y=Y-8:X=X-5:F$="THEY ARE CURSED":RETURN
1100 IFB$="3810010",GOSUB1370
1110 IF(A=15ANDO<>20ANDO<>1)OR(A=29ANDO<>16)ORO>C3,SL=1
```

```
1115 IFSL=1,F$=W$+C$+" "+X$:SL=0:RETURN
1120 IFL(O)=RAND(F(O)<1ORF(O)=9)ANDO<C3,L(O)=0:A=-1
1130 IFO=16ANDL(10)<>0,L(O)=R:F$="IT ESCAPED":A=0
1140 IFO>C1ANDO<C2,F=F+2:A=-1
1150 IFO>=C2ORO<=C3,G=G+2:A=-1
1160 IFO>C1ORO<C3,L(O)=-81
1170 IFA=-1,F$="TAKEN":X=X+4:E=E+1:IFF(O)>1,F(O)=0
1180 IFB$<>"246046"ORL(11)=0,RETURN
1190 F$=U$:L(O)=R:IFRND(3)<3,RETURN
1200 A$="#"+U$+R$
1210 R=63+RND(6):L(16)=1:F$=""
1220 GOSUB2740:RETURN
1230 GOSUB2770:F$="":A$="#THE LOGMEN "+M$
1240 F(41)=0:Y=Y-4:X=X-4
1250 IFR<34,A$=A$+"THROW YOU IN THE WATER":R=32
1260 IFR>33,A$=A$+"TIE YOU UP IN A STOREROOM":R=51
1270 GOSUB2750:GOSUB2760
1280 FORI=3TO4
1290 IFL(I)=0,L(I)=42
1300 NEXTI:RETURN
1310 A$="*THE SWAMPMAN TELLS HIS TALE"
1320 GOSUB2740:F(32)=-1:RETURN
1330 F$="MEDIAN CAN DISABLE THE EQUIPMENT"
1340 IFL(8)=0,F$=F$+" AND ASKS YOU FOR THE PEBBLE YOU CARRY"
1350 RETURN
1360 F(36)=-(RND(4)+6):F$="A STORM BREAKS OVERHEAD!":RETURN
1370 FORK=1TO30:GOSUB2770:PRINT"////LIGHTNING FLASHES!":NEXTK
1380 L(39)=R:Y=Y-8:X=X-2:RETURN
1390 IF(O<>24ANDL(O)>0)ORO=52,F$="YOU DON'T HAVE THE "+X$:RETURN
1400 PRINT"GIVE THE ";X$;" TO WHOM";:INPUTX$
1410 Q=O:GOSUB760:N=O:O=Q
1420 IFR<>L(N),F$="THE "+X$+" IS NOT HERE":RETURN
1430 IFB$="10045"ANDN=40,L(O)=81:F(40)=1:F$="THE SNAKE UNCURLS"
1440 IFB$="2413075"ANDN=30ANDG>1,SL=1
1445 IFSL=1,F(11)=0:F$="HE OFFERS HIS STAFF":SL=0:G=G-1
1450 B$=LEFT$(B$,3):F$="IT IS REFUSED"
1460 IFB$="300"ANDN=42,X=X+10:L(O)=81
1470 IFB$="120"ANDN=42,X=X+10:L(O)=81
1480 IFB$="40"ANDF(4)<0ANDN=32,F(N)=1:L(O)=81
1490 IFLEFT$(B$,2)="80"ANDN=43,L(O)=81:GOSUB1560
1500 IFL(O)=81OR(O=24ANDL(II)>0ANDG>0),F$="IT IS ACCEPTED"
1510 IFN=41,L(O)=51:F$="IT IS TAKEN"
1520 RETURN
1530 IFO=4ANDL(O)=0,L(O)=81:X=X-1:F$="IT BREAKS!":RETURN
1540 IFL(O)=0ANDO<=C1,L(O)=R:F$="DONE":E=E-1
1550 RETURN
1560 A$="*HE TAKES IT ":IFR<>8,A$=A$+"RUNS DOWN THE CORRIDOR,"
1570 GOSUB2740:A$="*AND CASTS IT INTO THE CHEMICAL "
1575 A$=A$+"VATS, PURIFYING THEM WITH A "
1580 A$=A$+"CLEAR BLUE LIGHT REACHING FAR "
1585 A$=A$+"INTO THE LAKES AND RIVERS BEYOND"
1590 F(8)=-1:GOSUB2750:GOSUB2760:GOSUB2760:RETURN
1600 IFL(I)<>0ANDI<C1,I=I+1:GOTO1600
1610 IFL(I)=0,L(I)=R:F(I)=0:GOSUB1540:F$="YOU DROP SOMETHING"
1620 RETURN
1630 IF(O<C1ORO>C3)ANDX$<>"???",F$=W$+C$+" "+X$:X=X-1:RETURN
1640 F$="YOU HAVE NO FOOD":IFF>0,F=F-1:Y=Y+10:F$="OK"
1650 IFO=3,X=X-5:Y=Y-2:F$="THEY MAKE YOU VERY ILL!"
1660 RETURN
1670 IFO=31,GOSUB2380:RETURN
1680 IFX$<>"???"AND(O<21ORO>C3),F$=W$+C$+" "+X$:X=X-1:RETURN
1690 F$="YOU HAVE NO DRINK":IFG>0,G=G-1:Y=Y+7:F$="OK"
1700 RETURN
1710 IFLEFT$(B$,4)="1600,F(O)=-1:F$="IT ALLOWS YOU TO RIDE"
```

Continued next issue.

128K S/WAYS RAM PT.II - J.LEON





THE ORIGINAL IDEA WAS TO PRODUCE A PRINTED CIRCUIT BOARD AND OFFER IT FOR SALE TO INTERESTED PERSONS. UNFORTUANATELY THE COST OF PRODUCING SUCH A BOARD IS WELL OVER $200.00 AND OUT OF MY REACH.

THE ALTERNATIVE IS TO USE EXISTING BOARDS WHICH CAN BE USED WITH SOME MODIFICATIONS. THE TWO BOARDS ARE :-

1) - ETI 687 SUPER II 34 K RAM BOARD.
2) - DAVID NEWCOMBE 32K RAM BOARD - SEE LAST ISSUE FOR DETAILS.

IN THIS ARTICLE I'LL BE USING DAVID'S BOARD AS AN EXAMPLE AND MOST OF THE IDEAS SHOULD APPLY TO THE ETI 687 BOARD AS WELL.

CORRECTION - IN PART 1 THE CIRCUIT OF IC 1 IS INCORRECT. THE CORRECT CIRCUIT APPEARS AT TOP LEFT OF THIS PAGE.

WHEN I WAS DESIGNING 128K S/WAYS RAM I ALSO DESIGNED CIRCUITS FOR 32K VIDEO RAM, 64K BASIC ROM, 128K DOS ROM, 128K RAM DISK AND 64K FOR 6000H-67FFH RANGE ALL USING S/WAYS OR BANK SWITCHING TECHNIQUES. INADVERTANTLY I USED THE WRONG CIRCUIT.

DAVIDS RAM BOARD IS DESIGNED TO HOLD 4 X 8K RAM CHIPS IF USED WITH VZ 200 OR 2 X 8K AND 1 X 2K RAM CHIPS PLUS A 74LS00 IF USED WITH VZ 300. TO USE WITH VZ 300 ONLY 98K (3 X 32K + 1 X 2K) CAN BE PLACED ON BOARD AND WITH VZ 200 80K (2 X 8K + 2 X 32K). THE MISSING 32K CHIPS COULD BE PIGGYBACKED OR PUT ON DAVIDS 64K ROM BOARD WHICH CAN HOLD UP TO 4 X 16K EPROMS.

THERE ARE ONLY 2 DIFFERENCES BETWEEN 8K AND 32K RAM CHIPS AND ARE MARKED WITH AN ASTERICK ON THE PINOUTS ABOVE. THEY ARE :-

PIN 1 8K +5V - 32K A14
PIN 26 8K +5V - 32K A13

WHAT IT MEANS IS THAT WE CAN PUT 32K RAM CHIPS IN PLACE OF 8K RAM CHIPS ON THE BOARD. PINS 1 AND 26 MUST BE BENT UP 90 DEG. OR MORE PRIOR TO INSERTING INTO SOCKETS.

128K S/WAYS RAM PT.II CONT

ONE HELPFULL FEATURE OF DAVID'S BOARD IS IT'S ABILITY TO TAKE A PIGGYBACK ROM BOARD WITH ALL THE REQUIRED SIGNAL LINES AVAILABLE BAR FOUR, RESET , A13, C000 AND E000. IC'S 1 TO 4 FROM 128K RAM CIRCUIT COULD BE PUT ON APIECE OF VEROBOARD AND UNIT PLUGGED IN WHERE THE ROM BOARD IS SUPPOSED TO GO.

THE DRAWING DEPICTING REVERSE SIDE OF BOARD SHOWS WHERE ALL THE REQUIRED SIGNAL LINES ARE. A13 CAN BE HAD FROM TWO LOCATIONS AND BOTH ARE MARKED. RESET IS AVAILABLE ONLY FROM EDGE CONNECTOR.

TWO MORE SIGNALS ARE NEEDED AND THEY ARE C000 AND E000 AND ARE MARKED WITH AN ASTERICK. THEY COME FROM ON BOARD DECODER WHICH DECODES THE FULL 64K IN 8K BLOCKS. THE DIODE AND GATE COMBINES THE TWO 8K BLOCK SIGNALS INTO ONE 16K BLOCK IN THE NEEDED RANGE OFF C000-FFFF WHICH ACTS AS AN ENABLE FOR IC 3.

ONCE YOU HAVE ASSEMBLED THE PLUG IN BOARD WE CAN PROCEED TOWARDS ASSEMBLY. IF USING WITH VZ 300 THEN THE 74LS00 DECODER COULD BE MOUNTED ON PLUG IN BOARD BY MAKING IT LONGER AND LETTING IT OVERHANG AT THE EDGE CONNECTOR END WHICH WOULD MAKE ROOM FOR THIRD 32K RAM CHIP ON RAM BOARD.

THE 32K RAM CHIPS ARE PLACED IN POSITIONS 4, 3 AND 2. THE CS (CHIP SELECT) SIGNALS COME FROM IC 4A TO 4B, PINS 3, 6 AND 11 AND GO TO CS POINTS MARKED ON 32K RAM BOARD. SOLDER A PIECE OF COVERED WIRE TO ONE OF A13 ADDRESS POINTS AND CONNECT OTHER END TO PIN 26 OF ALL 32K RAM CHIPS.

DO THE SAME WITH ADDRESS A14 DERIVED FROM PIN 10 OF IC2 AND SOLDER TO PIN 1 OF ALL 32K RAM CHIPS. THE 3 LEDS CAN BE INSTALLED NOW AND CAN BE REMOVED LATER IF DESIRED. THEY ARE HANDY IN TESTING CIRCUIT OUT. THE TRUTH TABLE FOR IC 3 CAN BE USED WITH THE LEDS TO TELL YOU WHICH BANK IS ACTIVATED.

PRESUMING ALL BUILT AND READY TO GO HOOK UP TO THE VZ LEAVING TOP OF CASE OFF POWER UP AND IF NO PUFF OF SMOKE PROCEED WITH TESTING UNIT. IF LEDS ARE CONNECTED THE LED CONNECTED TO D0, PIN 7 OF IC 2 SHOULD BE ON. USING LOGIC PROBE WE'LL TEST IC 3 OUT.

PIN 15 OF IC 3 SHOULD BE LO AND ALL OTHER O/P'S SHOULD BE HI, IF NOT THEN THERE'S AN ERROR IN ASSEMBLY. RESET VZ AND REPEAT AND IF OK PROCEED BY TESTING OTHER O/P'S ON IC 3 :-

OUT112,1 AND PIN 14 SHOULD BE LO, OTHERS HI - (BANK 1)
OUT112,2 AND PIN 13 SHOULD BE LO, OTHERS HI - (BANK 2)
OUT112,3 AND PIN 12 SHOULD BE LO, OTHERS HI - (BANK 3)
OUT112,4 AND PIN 11 SHOULD BE LO, OTHERS HI - (BANK 4)
OUT112,5 AND PIN 10 SHOULD BE LO, OTHERS HI - (BANK 5)
OUT112,6 AND PIN 9 SHOULD BE LO, OTHERS HI - (BANK 6)
OUT112,7 AND PIN 7 SHOULD BE LO, OTHERS HI - (BANK 7)
OUT112,0 AND PIN 15 SHOULD BE LO, OTHERS HI - (BANK 0)

IF EVERYTHING CHECKS OUT THEN IT'S TIME TO PUT IT TO USE AND WHAT IT'S USED FOR IS UP TO YOU. YOU COULD TRY WRITING A NUMBER IN EACH BANK CORRESPONDING TO BANK AS A FINAL TEST.

THE 128K S/WAYS RAM CAN BE EXPANDED TO 256K BY REPLACING IC 3 WITH A 74LS154, AN 1 OF 16 DECODER WHICH HAS 16 O/P'S. ANOTHER QUAD AND GATE WOULD HAVE TO BE USED AND 4 EXTRA 32K RAM CHIPS.

FLICKER-FREE GRAPHICS

by

Bob Kitch

PART 1

Mention the word "flicker" and some people immediately think of the horse in that well known TV series! VZ Users who have played DSE's Dawn Patrol know differently.

Flicker or hash on the VZ screen is that annoying "rubbish" that occurs, particularly on the hi-res screen, whenever anything is written to video RAM (VRAM). It is most often observed when graphics are being used. There are two causes of flicker in graphics -

i/ Critical Flicker Frequency as perceived by the eye. This is a form of "jumpiness" detected by the eye (i.e. physiological) whenever frame rates drop below 50 Hz or 50 per second. Movie theatre projectors and the video screen on the VZ are updated every 20 msec. and so smooth graphics result. Below this frequency, the eye sees a jerky image as is sometimes seen in old movies.

ii/ Video irregularities resulting from display design problems. Flicker or hash in the VZ results from the direct loading of VRAM from Z80 registers. The electron beam of the display scans every 20 msec. independantly of the program intentions. VRAM is memory mapped into 500/2K byte area depending upon mode. Conflicts in the timing of the Z80 CPU writing to - and the 6847 VDG reading from - the VRAM cause the flicker.

Consider the following example to understand the effect. The ASCII code for a character is loaded into VRAM from a Z80 register. The 6847 Video Display Generator (VDG) translates this into the intensity of the electron beam as it scans across the screen. The Z80 does not, and cannot, draw directly onto the screen. This is the job of the VDG chip. A screen is drawn every 20 msec. but the Z80 may load data into VRAM in a fraction of a msec. and at any time. Thus, the Z80 has now idea of where the electron beam is when it updates VRAM. The electron beam scans across the screen many times to form a complete character. As the VDG is updating the screen the Z80 may change the character in VRAM before it is completed by the VDG. (what a confusing mess for the screen to transfer to the eye!). A "split" section of screen scan lines may occur momentarily in this situation.

Ideally, all intended screen updates should take place at an "invisible memory location" or "buffer". At a suitable time, a Block Load instruction can move the buffer into VRAM. This transfer should occur AHEAD of the electron beam.

What is needed is some ability for the Z80 to determine the position of the electron beam on the screen at any given moment.

It is worth clarifying here, some of the hardware nomenclature used so that readers are quite clear as to what is going on. VRAM and buffer are areas of memory in the Z80's address space. The Z80 CPU can read and write (update) to these locations quite freely. VRAM in the VZ is a fixed area of memory that is also referenced by the 6847 VDG. The location of the buffer is chosen by the programmer and therefore the VDG does not know where it exists. Obviously the potential for conflict, between the two devices' requests for access to VRAM, exists here. The screen or display is a physical device that the Z80 knows nothing about. (the VZ will run without a screen). The VDG and a lot of display circuitry interface between VRAM and the image formed by an electron beam scanning across phosphors. The VDG contains an internal

ROM holding a shape table that decodes ASCII into visual images. Once these components are clearly distinguished, and their ability to communicate amongst themselves is appeciated, it becomes easy to devise a solution to their mutual interference that leads to flicker.

The Interrupt (as detailed in my previous article on vectors, interrupts and real time clock) is one such timing pulse that can assist. Every 20 msec. an interrupt signal is developed by the VDG. When the interrupt occurs, The Z80 completes the current instruction and then vectors to the Interrupt Handling Routine. After completion of this routine, the Z80 resumes its previous task. The interrupt pulse originates from the VDG and is notified by *FS line. This is connected to the Maskable Interrupt line on the Z80 which can be Enabled/Disabled by software commands. The important feature of the *FS line going low is that it allows video synchronization of the electron beam location on the screen with the Z80 write-to-VRAM activity. (i.e. exactly what we are looking for) By detecting the Maskable Interrupt event in the Z80 it is possible to time VRAM write-operations such that flicker is avoided. To see how this is achieved, it is necessary to consider how the screen scanning happens.

The VZ Technical Manual and my articles in Sept. and Oct. AEM 1986, provide some insight into screen scanning and the workings of the VDG. The electron beam starts at the top left-hand corner of the screen and moves to the right. In so doing, the VDG puts onto the display, part of 32 characters on the scan line.(this is of course describing the lo-res screen situation). During this time, the intensity of the beam is regulated by the VDG referencing VRAM. A "horizontal retrace" occurs that is invisible - also the beam drops down one scan line (this period corresponds to 8 "invisible" characters being scanned). There are actually 192 "visible" scan lines down the VZ screen. This corresponds to 12 scan lines per line of text. The VDG certainly earns its keep translating 32 ASCII representations per line into electron beam intensities!

So much for the visible or active portion of the screen. There are also "invisible" portions of the screen - for example the top and bottom borders of the screen. Also, once the electron beam reaches the bottom right-hand portion of the screen, it must "flyback" to the top left-hand corner. This is called the "vertical retrace". There is also another period called "blanking" which is a timing pause to allow the overworked electron beam to catch its breath before commencing another screen! Got all of that! In total, all of this beam activity corresponds to 312 scan lines - and it all occurs in 20 msec. or 50 times per second. For those whose maths can stand it, each scan line takes 0.064 msec. and each lo-res text line takes 0.77 msec. for the VDG to place upon the screen.

This set-up can be illustrated by the following summary -

50	lines	(3.21 msec.)	top border.
192	lines	(12.31 msec.)	active display.
51	lines	(3.27 msec.)	bottom border.
6	lines	(0.38 msec.)	vertical retrace.
13	lines	(0.83 msec.)	blanking.
---	-----	-----	-----
312	lines	(20.00 msec.)	

The *FS line of the VDG goes low (high to low transition) at the END of the active display area - i.e. before the bottom border is reached. The time when the electron beam is scanning "invisible" portions of the screen is marked by this signal, which, in turn,

Continued on next page.

triggers the Maskable Interrupt in the Z80. During this time, the CPU may have total access to VRAM without causing undesirable flicker on the screen. The amount of time available to the Z80 is 7.69 msec. out of the 20 msec. between each screen refresh. Note however, that the low to high transition of *FS coincides with the trailing edge of the vertical sync pulse - that is, it stays low for only 3.65 msec. or some 18% of the refresh cycle.

We now need to consider the timing of the CPU in doing a 2K Block Move operation. The Z80 in the VZ300 runs at 3.54 MHz (the VZ200 runs at 3.58 MHz). The following tabulation indicates the timing of the Block Move -

Opcode	T-states	M-cycles
DI	4	1
LD HL	10	3
LD DE	10	3
LD BC	10	3
LDIR	21x2047+16	5x2047+4
	If BC=0 then T=16 and M=4	
	If BC<>0 then T=21 and M=5	
EI	4	1
RET	10	3
	---	---
	43051	10253

Time to execute is 43051/3.54E6 sec. = 12.16 msec. to Block Move a 2K screen. This is longer than the 7.69 msec. mentioned above. A 2K Block Move will cause hash to appear on the screen because a timing conflict in the write by the Z80 and the read by the VDG occurs. For those of you who have seen THROWUP operate, you will have seen this effect in the "plain mode" of operation.

So, what have we learnt thus far about graphics on the VZ? We have discovered how the Z80 and the 6847 VDG interact with VRAM to make the screen display. We have also seen why "hash" appears on the screen. We have also analysed the timing of screen transfers from a buffer and discovered that about 1K of data can be Block Moved during the screen refresh period. In some previous articles in User Group Newsletters, I have described the Maskable Interrupt System used in the VZ. Let's put all of this together to make a hash-free display system. Such a system is used in LIVENUP.

Continued next issue.

P.O BOX 1972 C.P.O AUCKLAND N.Z

AUCKLAND VZ 300/200 COMPUTER USERS CLUB (1986)
NEW ZEALAND VZ 300/200 COMPUTER USERS CLUB (1989)

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT.
CONTRIBUTED PROGRAMS AND ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT IT IS FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.

COPYRIGHT IS ALWAYS RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR ANY BREACHING OF COPYRIGHT CAUSED BY THE PUBLICATION OR USE OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME ON.....PHONE 520-6743

USING DISK TOKENS (C) R.QUINN

USING A DISK TOKEN IN A BASIC PROGRAM :-

First we choose a disk token--133 will do. Then we insert this token in a brief BASIC program to test its operation. Enter the following program:-

```
10 LET:PRINT"TEST"
20 PRINT"AGAIN"
30 END
```

To replace the token for LET with our disk token,
POKE31469,133

If you now RUN you will get ERROR messages because your VZ's MICROPROCESSOR (the MP) picked up on token 133 at start of line 10, located the pointer for this token which directed it to address 31091 where it found and executed a JP301 instruction.

What we want to do is change the instruction at 31091 so that the MP will execute it, then return to line 10 and continue to run our BASIC program. The simplest change I know of is a RET instruction (Code 201). So, POKE31091,201 and RUN.

It works! The MP jumped to 31091 where the RETurn was encountered and executed and program RUN continued on to complete line 10 and then line 20. This tells us something important: when a token is encountered in a BASIC program, if a CALL instruction is executed, so that no matter where the MP may jump to run the M/L routine tied to that token, it can find its way back to its position in the current BASIC line if the M/L routine ends with a RET.

So let's try something a little more complicated. We'll set up a short M/L routine in a stretch of otherwise unused memory, in the COMMUNICATIONS REGION, that begins at address 31273 (LO=41; HI=122). First change the instruction at 31091 to JP 31273:

POKE31091,195:POKE31092,41:POKE 31093,122

Now add these lines to our BASIC program:

```
50 A=31273
60 INPUTB:POKEA,B:A=A+1:GOTO60
```

RUN50 and INPUT this sequence of numbers: 33,1,1,34,0,112,201

BREAK -- We have now set up a M/L routine at 31273 consisting of these three instructions:-

```
LD HL,257
LD(28672),HL
RET
```

The routine loads the H and L registers with '1' which is the POKE code for A, then POKEs these codes to the first two cells of screen memory (28672/3, top left corner) and RETurns.

So, now CLS, press <RETURN> and RUN.

?SYNTAX ERROR IN 10

USING DISK TOKENS CONT. . . .

But our M/L routine executed, did it not? There are two A's in top of screen. Only the MP returned to line 10 and BREAKed. Why? Because we changed the contents of the HL registers. When we use USR(x) to execute a M/L routine, USR saves the current contents of the HL registers on the STACK and when RET is encountered to bring the MP back to BASIC, the contents of the hl registers are restored. No matter how the registers may change in the course of executing the M/L routine, the MP continues execution of the BASIC program with the registers the same as when it began USR.

We must do the same with our M/L routine. The routine only uses the HL register pair so we need only save and restore these two registers. We do this by PUSHing HL on the STACK at the start of our routine and then POPing them from the STACK at the end of the routine.

```
229      PUSH HL
225      POP HL
```

Again RUN50 and INPUT this sequence of numbers: 229, 33, 1, 1, 34, 0, 112, 225, 201

BREAK. CLS, press <RETURN> and RUN.

This time our program ran through to END in line 30, executing our M/L routine, then returning to execute the two PRINT statements in lines 10 and 20.

Having successfully worked out a technique for making use of disk tokens, it only remains to design some interesting and useful M/L routines to execute with our new technique. That will be largely up to you. But here is an example:-

Video memory consists of 2048 bytes (from 28672 to 30719), of which only one quarter (512 bytes) is used for the screen display in text mode (28672 to 29183). We can use another block of 512 bytes of video memory as a video store, say from 29184 to 29695, and set up M/L routines to copy the text mode screen to our video store and recall the content of the video store back to screen. This can be done very easily entirely in BASIC, using PEEK and POKE in FOR NEXT loops. The advantage of M/L routines is that they are very fast, practically instantaneous.

Again RUN50 and INPUT the following sequence of numbers :-
229,33,0,112,17,0,114,1,0,2,237,176,225,201
229,33,0,114,17,0,112,1,0,2,237,176,225,201

And BREAK. Two routines have been set up in the COMMUNICATIONS REGION, a COPY (screen to video store) routine starting at 31273 and a RECALL (from video store to screen) routine starting at 31287.

We will use the disk token 165 to execute the COPY routine (PUT) and disk token 164 to execute the RECALL routine (GET). We must change the JP instructions in the COMMUNICATIONS REGION for these disk tokens to make them jumps to the start of our M/L routines:

```
POKE31107,41:POKE31108,122    POKE31104,55:POKE31105,122
```

We are now ready to use our new tokens and their M/L routines in BASIC programs. So let's try them out. First NEW your VZ.

USING DISK TOKENS CONT. . . .

Enter this BASIC line :-
500 LET:RETURN
Now POKE31469,165 and enter this line :-
400 LET:RETURN
And POKE31469,164

We have set up two BASIC subroutines, a PUT routine at line 500 and a GET routine at line 400 which can now be called by using GOSUBs to those lines from within our BASIC program or even called with GOSUBs in command mode.

Try it. Enter GOSUB500
NOW CLS and GOSUB400

How about that eh? You can fill the screen with any data you please, then store it in the video store with a GOSUB500, and recall it to screen anytime you want with a GOSUB400.

****
HELPFUL HINT #1 "INIT"

Did you know that it is faster when FORMATTING a number of disks to switch backwards and forwards between TWO DISK DRIVES?

```
INIT:DRIVE2:INIT:DRIVE1:INIT:DRIVE2:INIT:DRIVE1:INIT
```

OR

```
10 INIT:DRIVE2:INIT:DRIVE1:GOTO10
```

Try it out, changing disks as each drive INIT's a disk, and appreciate the greater efficiency.

****

*** ELECTRONIC INDEX ***

Volume I & II holds every item ever printed in your VZ LINK NEWSLETTERS from the beginning right up to the end of 1990.

Your copies are available now.

$4.50 each ..... DISK or TAPE

* FREE DATABASE PROGRAMME
* FREE DISK OR TAPE
* FREE UPDATING SERVICE
* FREE DATABASE MANUAL

Volume III will hold more entries. The database allows you to do this yourself, month by month. Alternatively, you are invited to take advantage of the free updating service. You are welcome to have your original Disk/Tape up-dated at no charge.

[Please always remember to provide return postage and do use good packing to prevent damage in the post.]